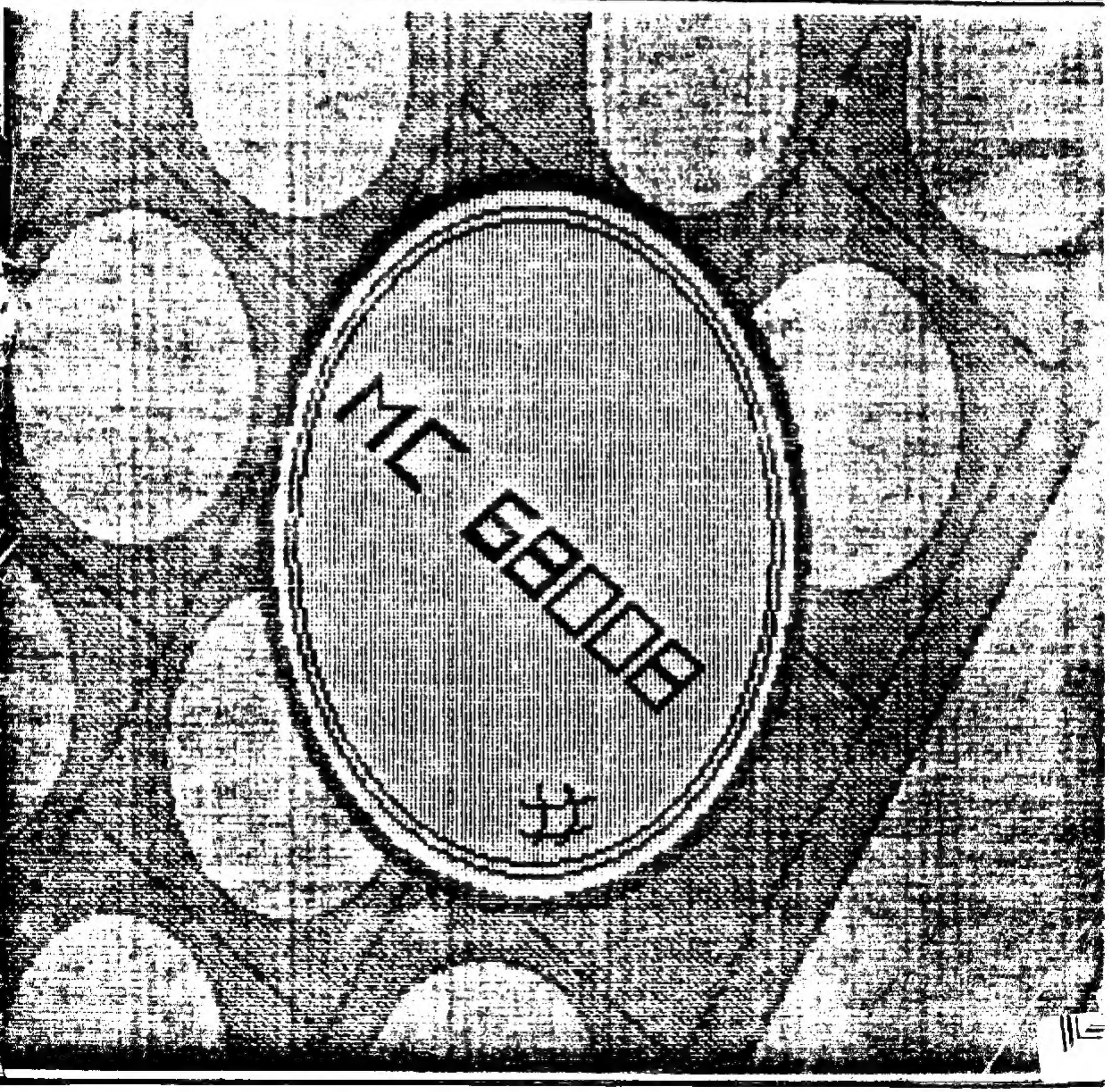
COMPUTER

VOLUMEN I NO. 1 ENERO 1986



INFORMACION SOBRE EL CLUB

La integraciòn en la asociaciòn C.E.I.U.Q.L. se hace por suscripciòn anual. El C.E.I.U.Q.L. publica mensualmente la revista de los socios QLave, nombre por el cual tambièn se conoce al club.

Màs informaciòn sobre la asociaciòn se puede obtener desde la secretaria del club.

Para ser miembro de QLave se requiere estar interesado en el micro-ordenador Sinclair QL.

El club mantiene una libreria de software. Una lista de los programas existentes en la libreria se publicarà de vez en cuando para asi ir actualizàndola. Los programas que se quieran aportar o sacar de la libreria se deben notificar directamente al encargado de la misma.

Por favor enviad todas las contribuciones a QLave al presidente.

Presidente	Secretario	Librero
Serafin Olcoz	Juan Palacio	Angel Asin
Baltasar Graciàn 21,1-cto.	Pedro Mª. Ric 19, 4	San Jorge 22,3-B
50005 ZARAGOZA	50008 ZARAGOZA	50001 ZARAGOZA

Contribuciones a QLave:

Las contribuciones a QLave deben ser Archivos de QUILL en cartuchos de microdrive, preferiblemente. Los cartuchos se devuelven a vuelta de correo.

Los programas cuya extensiòn no sea muy grande se incluiràn en la revista, pero aquellos de gran extensiòn pasaràn a formar parte de la libreria.

Los programas que enviàis deben adjuntar una descripciòn de los mismos y de su funcionamiento.

========== INDICE ==========

- Portada
- Informaciòn sobre el Club
- Manual de referencia del Sinclair QL
- ALSIMAIL
- QL CHESS (PSION)
- Lenguaje màquina para el QL (I)
- QUANTA
- Trucos,Rutinas ...

Se prohibe la reproducciòn total o parcial del contenido de esta revista.

Sinclair, QL, QDOS, ZX microdrive son marca registrada de Sinclair Research Ltd.
QUILL, ABACUA, ARCHIVE, EASEL son marca registrada de PSION Ltd.

EDITORIAL

Aquí está ya el número uno de la revista QLave, TU REVISTA.

Como podeis observar en este número está todo lo que de él avanzábamos en el número de presentación con la excepción de que todavía no hemos abierto la sección de cartas de los socios, que a mi parecer deberá ser la columna vertebral de QLave. Espero que ésto sea un toque de atención a todos los socios y que pronto os animéis a colaborar y a dar forma a esta revista.

También hemos modificado el contenido de la sección Rincón del Código máquina. Ya que deberíamos inaugurarla, según el avance del número 0, con una introducción a la arquitectura del microprocesador 68008. Pero hemos pensado hacer antes una introducción más general y dejar la de la arquitectura para próximos números.

La finalidad de esta modificación es abrir más el abanico para que así se puedan 'enganchar' también los principiantes que sean legos en la materia. Espero que aquellos que tengan ésto más que trillado sepan comprenderlo.

De todas formas y como os decía en la presentación espero que me expongáis qué tipo de secciones y de artículos os gustaría que apareciesen y si os parece bajo o normal el nivel de los mismos. Así, indicándonos los errores, nos iremos corrigiendo poco a poco.

Aunque en el número 0 (Presentación), en la sección dedicada a información sobre el club incluíamos las direcciones y cargos de los socios fundadores del C.E.I.U.Q.L., hemos pensado publicar a partir de este mismo número solamente los correspondientes al presidente, secretario y librero; ya que son a quienes debeis remitir vuestras colaboraciones a QLave, petición de información e inscripción y todo lo relacionado con la librería, respectivamente.

Con respecto a las colaboraciones que me envieis para publicar en la revista QLave, debéis indicarme si queréis que ponga además de vuestros nombres, apellidos y direcciones y si lo hubiera vuestro teléfono. Ya que si no me indicáis nada simplemente pondré vuestros nombres y apellidos.

Debo hacer una advertencia a los piratas de software y es que cualquier miembro de QLave del que se tenga conocimiento que realice tal actividad será automáticamente expulsado del club y perderá por consiguiente todos sus derechos sobre el mismo.

Esta medida, recogida en los estatutos del club, se toma para proteger la propiedad intelectual de los programadores así como los derechos de las casas comerciales de software.

Las portadas de este número y del anterior han sido realizadas por nuestro tesorero: Isidro Asín con ayuda del programa GRAPHIQL. Como esta labor lleva bastante tiempo os lo comunico por si alguien quiere colaborar enviando portadas (grabadas en un cartucho).

Por último tengo que deciros que todavía no hemos tenido ninguna noticia del Gobierno Civil referente a los estatutos, aunque todavía es pronto ya que su contestación debe llegar en el plazo de un mes y éste aún no se ha cumplido.

Serafín Olcoz

MANUAL DE REFERENCIA PARA EL SINCLAIR QL
Autor: TIM HARTNELL

Se trata, como su propio nombre indica, de un manual para el QL que sobre todo tenia gran utilidad cuando en España solamente se comercializaba el QL en versión original y con su manual en inglès, por supuesto.

Se puede decir que los siguientes capitulos son casi una mera traducción de la guia del usuario, pero eso si, incluyendo en algunos casos varios ejemplos interesantes:

1- Primeros pasos en el QL, 2- Cuidado y carga de los microdrives, 3- Conceptos del Superbasic, 4- Identificadores, 5- Operadores, 6- Cadenas, 7- Matrices, 8- Primeros pasos en la programación estructurada, 9- Funciones y procedimientos, 10- Gràficos, 11- Sistemas de coordenadas, 14- Ventanas, 16- Tratando con DATA, 17- Extendiendo su vocabulario, 18- Manejo de Ficheros.

Asi como los apèndices :

1- Demostración de multitarea, 2- Funciones matemàticas, 4- Una introducción al 68008, 5- Mensajes de error, 8- Conexionado.

Como novedades respecto a la guia del usuario cabe destacar el capitulo: 12- Gràficos definidos por el usuario. Capitulo formado por el programa de Derek Wilson, publicado en la revista QUANTA, que define los gràficos asi como por algunos comentarios y ejemplos que completan el capitulo.

También se puede destacar el capitulo 13- QLogo, en el que se aprovechan los comandos definidos en el Superbasic encargados de realizar gràficos de tortuga caracteristicos del Logo, para hacer el programa QLogo que emula en gran parte, aunque por supuesto no del todo, el lenguaje Logo.

Asi mismo el capitulo 15- El sonido y la música, tienen la caracteristica frente al manual de Investrònica(Sinclair) de no limitarse simplemente a definir el comando BEEP e invitar a realizar tus propias experiencias sino que lo explica algo mejor e introduce algunos ejemplos muy didàcticos.

Respecto al capitulo 19- Modelo Financiero, hay que decir que està bien como ejemplo de programa en Superbasic, pero que realmente carece de utilidad frente a los programas de PSION que se suministran con el propio QL.

El capitulo 20- Perfeccionando sus programas, hace una introducción a los diagramas de flujo asi como una referencia muy loable a las ventajas de escribir los programas en papel antes de pasarlos al QL.

En el capitulo 21- La màquina FORTH, se hace una introducción al lenguaje FORTH y al notación Polaca inversa y asi mismo se construye un programa, la màquina FORTH, que simula parte de este potente lenguaje.

Los dos últimos capitulos 22- Simulando la realidad y 23- Creando aventuras, ponen los ladrillos bàsicos para entender y realizar, al menos en esquema, los denominados 'juegos de parrilla' del tipo conversacional. En concreto el capitulo 23 està formado por dos juegos de ese tipo bastante parecidos.

El apéndice 3- Todo empezó todo, no es sino la reproducción de la invitación de Sir Clive Sinclair a la presentación, en una conferencia de prensa, del QL el 12 de Enero de 1984.

El apéndice 5- Glosario de términos informáticos, no necesita comentarios.

El apéndice 7- El Grupo Independiente de Usuarios del QL, ahora denominado QUANTA, como su revista , ya se [illegible] hace una referencia a dicho grupo al que como ya se ha dicho anteriormente pertenecen varios de los programas ejemplo que aparecen en este libro.

Por último el apéndice 9- Lecturas complementarias, cita seis libros que pueden ser interesantes.

La versión en castellano de este libro fue publicada por la editorial RA-MA en Febrero de 1985, su precio de venta al público es de 2400 pts.

Serafín Olcoz.

COMENTANDO EL ALSIMAIL

El ALSIMAIL es un programa creado por ALSI Comercial S.A. que sirve para crear ficheros de direcciones, recibos mensuales y mailing.

Su capacidad máxima es de 800 fichas por fichero y cartucho.

La memoria libre dejada por el programa tras su carga es de 20992 bytes, conforme se opera con el programa éste actualiza en la zona de mensajes la memoria libre que nos resta.

La pantalla aparece dividida en dos secciones:

- Zona de mensajes:

En la que se indica: "No pulsar nunca CTRL+SPACE", para volver al menú "0", para detener listados "CTRL+F5", num. de clientes, memoria libre y que al final hay que actualizar siempre los ficheros "Opción 0".

- Pantalla principal:

Se trata de la pantalla de trabajo en la que aparecerán las opciones e introduciremos los datos.

Al comenzar a trabajar es preciso tener siempre además del ALSIMAIL en el mdv1, otro formateado en el mdv2. Ambos cartuchos deben permanecer todo el tiempo que se trabaje con este programa en sus respectivos lugares ya que el programa accede a ellos en diversos momentos durante el periodo de trabajo.

El menú principal consta de las opciones habituales en este tipo de programas, abrir y borrar ficheros, cargar datos e impresión.

La opción de impresión simplemente nos pide la velocidad de transmisión, en baudios, a los que vamos a transmitir.

la de abrir ficheros pide nombre de éste y longitud de los campos(nombre, clave, calle, ciudad, provincia, otros). El núm. máx es de 10 ficheros diferentes por cartucho.

La de cargar datos es la más extensa con el siguiente menú:

1- Obtención de listados:
Puede ser de Clientes, Totales y vuelta al menú.
Para acceder a la opción de totales es precisa haber introducido con anterioridad los datos referentes al ejercicio en curso , posteriormente haber actualizado (opción 0) los ficheros y por último entrar en esta opción de totales. Se trata pues de una opción 'resúmen' clasificada por meses.
Respecto a la opción de clientes se nos presentan las siguientes opciones : Listados, por un concepto, por varios conceptos y vuelta al menu.
Así mismo hay que elegir entre: todos los campos o sólo algunos; entre: en línea, o mailing y por último elegir entre pantalla e impresora.
2- Introducción de datos:
Se nos van pidiendo los datos a introducir para cada uno de los campos antes definidos en la opción de "abrir ficheros". Cuando deseemos salir de esta opción basta con pulsar "*".
3- Impresión:
Hay que elegir entre: todos, un concepto, varios conceptos y vuelta al menú.
Posteriormente entre pantalla e impresión.
4- Bùsqueda, borrado, impresión y recibos.
5- Creación de recibos:
Indicando Localidad y fecha de expedición así como el importe del recibo.
6- Borrado y reapertura de ficheros totales.
7- Nos informa de los campos definidos.
8- Cambio de fichero.
0- Actualizaciòn y copias.

Así pues se trata de un programa realmente interesante para el profesional, simplificando su trabajo.

Los menus y opciones incluidas son a nuestra manera de ver bastante completas, para el uso que se le va a dar.

Respecto a las opciones de impresión decir que excepto en la opción de listados en línea que si incluye en software el código de salto de línea, en el resto de las opciones esto no sucede por lo que cada cual deberà adaptar su impresora para ello. En particular para la impresora admate 100 que carece de un interruptor para salto de línea automàtico y para otras impresoras que se encuentren en la misma situación ALSI Comercial S.A. recomienda hacer un puente entre las dos medias lunas que se pueden observar en la tarjeta de la impresora para resolver el problema.

Entre las diferentes clases de impresión para las que se puede emplear este programa podemos destacar:

Impresión de recibos modelo estàndar a una ficha determinada.
Impresión de todo el listado o de las fichas seleccionadas en papel a 80 columnas para règimen interno.
Impresión de todo el listado o de las fichas seleccionadas en formato de etiqueta adhesiva para envío de circulares.
Impresión en etiqueta de una ficha determinada repetidamente.

Su precio es de 18.000 pts. en microdrive o 20.000 pts. en disco.

Serafín Olcoz
Isidro Asín

COMENTARIO DEL QL-CHESS (PSION)

Realmente no es fàcil realizar un comentario de un programa sobre el que pràcticamente està todo dicho.

Hablar del QL-Chess es realmente decir ajedrez.

Aquí desde luego no vamos a descubrir nada nuevo sobre sus cualidades técnicas. Fue proclamado por méritos propios campeón del mundo de microcomputadores en cuestión de ajedrez. Su visión tridimensional del tablero así como de las piezas nos dan muestra de su gran atractivo, posee así mismo una serie de comandos que nos hacen de èl un gran compañero y elemento de ayuda.

Constan en su memoria 4.000 jugadas teóricas así como una serie de 28 niveles de dificultad, pudiendo escoger el que màs nos convenga o incluso variar su nivel a lo largo de una partida; posteriormente a todos estos aparecen opciones para lograr en una posición determinada dar mate desde una a ocho jugadas -muy útil para cualquier jugador de ajedrez que posea una partida aplazada, o incluso para resolver problemas complicados-. Una característica sobresaliente y que me llamo la atención sobre manera es la utilización de dos relojes que controlan, con exactitud sorprendente, el tiempo invertido, tanto de uno como de otro jugador, a lo largo de la partida.

Para poder realizar ester artículo QLave ha requerido la ayuda de D. Cristobal Ramo, excelente ajedrecista que ha ostentado el titulo de campeón de Zaragoza.

Después de haber estado delante del programa cinco horas, analizàndolo y compitiendo con èl, le hacíamos una serie de preguntas a modo de entrevista.

- Cristobal, a tu modo de ver ¿què papel cumplen hoy en dia las màquinas y ordenadores en el ajedrez?

- Bueno, actualmente se puede decir que los ordenadores cumplen una función màs que nada de marketing, es decir para venderlos. Es imposible y no se da el hecho de que los computadores sirvan de entrenadores, sin decir que màs adelente sirvan para ello. Actualmente los programas son preparados por jugadores màs bien de alto nivel y en un buen equipo pero todavìan se encuentran en un lugar un poco bajo si bien se desarrollarán mucho màs.

- Y esto ¿cuanto tiempo puede tardar?.

- Pues, dentro de un futuro màs o menos próximo quizàs llegue a tardar cinco o diez años, en este tiempo ya podràn cumplir una función, no de participar en torneos, humanos me refiero, pues aquí la condición de una màquina seria muy distinta a la de cualquier persona, sino digamos como una especie de fichero o de preparación de jugadores de alto nivel para perfeccionar un poco su juego en concreto realizar la función que hace todo entrenador, preocuparse por la apertura que va a jugar el dia después o la preparación de un match, etc...

- ¿Qué te ha parecido el GL-Chess en general, como programa?

- Verdaderamente me ha sorprendido gratamente, pues la última vez que jugué contra un ordenador no alcanzaba este nivel de juego, incluso hace unos años era decepcionante pues realizaba las cuatro o cinco primeras jugada bien y luego decaía estrepitosamente. Aquí ya se ve un perfeccionamiento muy acentuado principalmente en aperturas y un nivel de juego, diríamos, superior a la media actual en una ciudad como Zaragoza.

- En concreto el GL-Chess tendría alguna opción participando en torneos contra seres humanos?

- Dependería de la situación en que se desarrollase el torneo, y en concreto la categoría. A nivel local y adaptándose al sistema de juego es decir dos horas y media para realizar cuarenta movimientos las posibilidades no son del todo malas, creo que estaría a un nivel aproximado, sin exagerar, de categoría preferente. Con esto quiero decir que si participara en una categoría de segunda, que es la más inferior, ganaría sin dificultad; si estuviera en primera lograría ascender a preferente, que es una categoría más elevada, pero probablemente no se quedaría el primero; y una vez en preferente estaría, lo más seguro, bien clasificada dando una talla normal. Ahora bien a mayor nivel ya sería un poco más difícil podríamos calificar este programa basándonos en el criterio de selección de los jugadores o en el criterio de su clasificación que es el sistema ELO creo que tendría una puntuación próxima a los 2.150 que a nivel provincial en Zaragoza hay ocho o nueve jugadores que la alcanzan pero a nivel nacional evidentemente hay bastantes más es decir que obtendría un nivel medio dentro de un ámbito nacional; posteriormente a nivel internacional hoy todavía es muy pronto para que se enfrentara en torneos más o menos importantes.

- ¿Es posible que un ordenador llegue a superar a la mente?

- En principio lo máximo que podría hacer sería igualarla, puesto que el que programa es un ser humano y por lógica una mente. Lo que ocurre es que el hombre se encuentra en unas situaciones de desventaja y ventaja en una partida de ajedrez, puesto que influyen muchos factores externos a la pura objetividad que pueda tener un programa es decir que en una partida de ajedrez influyen factores tan normales como puedan ser los nervios, o el estado anímico de esa persona en ese instante, cosa que no le ocurre a la máquina, eso sí yo le veo al ordenador que siempre le faltará esa chispa de intuición, de reflejos en una posición determinada. Pero lo que está claro sin duda es que puede llegar a equipararse con el hombre incluso vencerle en más de una vez, pongo como exponente de este hombre al campeón del mundo, pero como ya digo ahora todavía es muy pronto para esto, incluso creo que pasarán bastantes años antes de que se pueda conseguir.

- Pasando ahora al programa en sí, de todos es sabido que una partida por regla general consta de una apertura, de un medio juego, y de un final: ¿Cómo ves al GL-Chess en aperturas?.

- En aperturas creo que puede ser el aspecto en el que se vea un mayor avance en este sistema. Es evidente que al poder ser esto matematizado mucho màs fàcilmente que el medio juego y que la mayoria de los finales que ya pueden ser complejos y pueden llegar a dar lugar a infinitas posiciones, en la apertura se parte de cero es decir es fàcil poder meterle los datos al programa. Suele seguir bastante bien la teoria en general entonces hasta la jugadaa diez o doce o incluso en que aperturas sean hasta la jugada veinte suele llegar a posiciones muy cèrcdas para ella. Resumiendo yo creo que en aperturas si hubiera que darle una calificaciòn desde luego habria que darle un sobresaliente.

- ¿Y en cuestiòn de medio juego?

- En esta fase de juego las dificultades aumentan pero hay que hacer notar que las ideas que tiene son bastante buenas fundamentalmente agresivas y quizà descuide un poco la reacciòn que pueda tener el contrario, es decir se preocupa un poco màs de atacar los puntos dèbiles del enemigo descuidando en parte su propia defensa. En general el nivel es aceptable sino es tan bueno como el que tiene en aperturas.

- ¿Con respecto a los finales de juego?

- En finales todo depende de la mayor o menor sencillez que tengan estos. En los finales un poco complejos pues se desenvuelve en un nivel de juego digamos parecido al del medio juego es decir descuida un poco su defensa y se dedica màs bien a buscar las debilidades del contrario. En finales tipicos ya cuando hay mayor simplificaciòn igual que la apertura son finales que pueden darse a un anàlisis màs exaustivo y por lógica que pueda adaptarse mejor a ellos la màquina. En resumen diria que es una variaciòn según el tipo de finales o sea contra màs complejos sean el ordenador vè una mayor dificultad porque los finales sobre todo es la fase de partida que hay que trataria con mayor sencillez, entonces segùn los casos el programa rechaza una jugada pasiva aunque sea la mejor por intentar buscar a su vez un contrataque, y ese quiza sea su mayor defecto.

- ¿Què piensas de la valoraciòn que te ofrece el ordenador durante una partida?

- Esto es algo que se puede hacer en cualquier momento con esta màquina y valora su posiciòn con un margen de error muy pequeño, quizà podriamos decir que es un poco optimista en este aspecto puesto que valora un tanto positivamente la posiciòn a su favor, que esto concuerda con lo que hablamos hablado antes sobre su forma de jugar que la hace responder de una forma agresiva.

- ¿Cual es la mejor virtud que has encontrado en él?

- La mejor virtud que he podido experimentar es su gran espiritu de lucha, puesto que nunca da una partida ni una posiciòn por perdida.

Desde luego, y ya para terminar diria que perfectamente a mi nivel me podria servir de entrenador.

Angel Asln

LENGUAJE MAQUINA PARA EL QL (I)

Comienza una sección que irá apareciendo en números sucesivos, en la que os ofrecemos una introducción al microprocesador 68008 y al código máquina del QL.

Para programar con el QL en lenguaje máquina, tan importante es conocer el 68008 como el sistema operativo del ordenador: el QDOS. En estas páginas trataremos de exponer sólo la primera parte, esto es, todo lo referente al 68008, de manera que cualquiera que haya seguido estos articulos, esté en condiciones de poder usar un programa ensamblador. Sólo restará para poder hacer rutinas en código máquina, conocer al fondo el QDOS, que aunque seguramente se tratara en números posteriores, se encuentra perfectamente explicado en un libro de reciente aparición: "PROGRAMACION AVANZADA PARA EL QL" de la editorial RA-MA.

Resulta dificil sin embargo encontrar bibliografia castellana sobre el 68008, por eso vamos a intentar desde aqui la ofrecer para todos los socios la información necesaria para poder empezar a hacer pinitos en CODIGO MAQUINA.

1.- QUIEN ES EL 68008

El 68008 es un microprocesador de la familia del 68000 diseñado y fabricado por MOTOROLA, No obstante en la actualidad tienen licencia para la fabricación de este microprocesador: Hitachi, Mostek, Rockwell, Signetics y Thompson. Esta familia está formada por circuitos integrados tales como: 68008, 68010, 68020, 68200, 68120, 68121, 68451, 68454...

El 68008 está clasificado como un microprocesador de 16 bits. Esto significa que trabaja con una longitud de palabra de 16 bits. El Spectrum, por ejemplo usa el microprocesador Z80, catalogado como de 8 bits por ser esta la longitud de datos que puede manipular de una vez.

El 68008 presenta la misma arquitectura que el 68000, basada en registros de 32 bits, y brinda además la posibilidad de trabajar con un microprocesador de 8 bits, superando en este punto a la mayoria de los microprocesadores de su categoria.

El software es completamente compatible con el del 68000, disponiendo de las 56 instrucciones y los 14 modos de direccionamiento de dicho microprocesador. (Actualmente en el mercado sólo existe bibliografia referente al 68000, pero afortunadamente el juego de instrucciones es exactamente igual al del 68008).

El aspecto externo del 68008 se reduce a una cápsula con 48 patillas, a través de las cuales el QL recibe y envia los impulsos eléctricos que le relacionan con el resto del QL.

2.- INTRODUCCION

El cerebro del QL está formado por un potente microprocesador: el 68008. Dicho microprocesador es el encargado de soportar y dirigir todo el trabajo de la máquina, pero sólo es capaz de entender instrucciones escritas en su propio lenguaje: el código máquina.

Todos los procesos que ocurren en el QL se reducen en última instancia a impulsos eléctricos o a una ausencia de los mismos que recorren la circuitería de la máquina. Así pues el 68008 es capaz de realizar distintos procesos según las combinaciones de impulsos que lleguen hasta sus patas.

Si hay señal en una de las patas del microprocesador escribiremos un 1 y un 0 si no la hay. Una combinación de 16 señales se representa tal que: 1001110011110101 y representa una instrucción para el 68008 que desde luego no se parece en nada a instrucciones como: IF A=3 THEN...

El QL tiene un programa llamado "intérprete BASIC", encargado de traducir un programa en SuperBASIC a rutinas en código máquina comprensibles para el 68008. Este programa intérprete, junto con todas las rutinas necesarias para el funcionamiento del QL (lectura del teclado, refresco de la imagen del monitor, funcionamiento de la NET...) está ubicado en la ROM (Read Only Memory -memoria sólo de lectura-).

La memoria del QL la podemos dividir en:

RAM: todas las celdillas de memoria en las que se pueden almacenar datos y leerlos. En el QL hay unas rutinas (gestor de memoria) encargadas de distribuir y controlar la RAM, para almacenar los programas BASIC, rutinas del usuario...
ROM: Zona de la memoria de la que únicamente se pueden leer los datos almacenados Esta zona va grabada desde la fabricación del ordenador y es inalterable. Tampoco se pierden los datos almacenados al desconectar el QL como le ocurre a la RAM.

Las unidades para medir la información son:

-bit: dígito binario (sólo admite dos posiciones 1 ó 0)
-byte (u octeto): combinación de 8 bits
-palabra: combinación de 16 bits
-palabra larga: combinación de 32 bits
-Kilobyte: múltiplo del byte formado por 1024 octetos.

La unidad mínima de información es el bit. Un bit sólo puede presentar dos aspectos: 1 ó 0 que se corresponden en los circuitos del QL con la presencia de una señal eléctrica (1) o la ausencia de la misma (0).

Todas las operaciones aritméticas que realiza el ordenador se llevan a cabo en aritmética binaria.

3.- SISTEMAS DE NUMERACION

1. Sistema decimal.

Utiliza 10 dígitos para expresar cualquier cantidad, teniendo en cuenta que la posición de cada uno le da un valor determinado, correspondiente a las distintas potencias de 10.
Por ejemplo el valor del número 1424 se obtiene sumando los valores de cada dígito según el lugar que ocupa:

$$1424 = 1*10^3 + 4*10^2 + 2*10^1 + 4*10^0$$

Así, el dígito de la derecha representa las unidades, el siguiente las decenas, el otro las centenas...

2. Sistema binario.

Se basa en la utilización de dos dígitos para expresar cualquier magnitud. Para obtener el valor de cualquier número en binario, se sigue el mismo procedimiento que con el sistema decimal, teniendo en cuenta que en lugar de trabajar con potencias de 10, hay que trabajar con potencias de 2.

$$1010_2 = 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = 8+0+2+0 = 10_{10}$$

A continuación se relacionan las equivalencias entre los primeros números decimales y los binarios correspondientes:

decimal	binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

- De binario a decimal

Para convertir un número en binario a decimal basta con realizar la operación anterior, multiplicando cada dígito por 2 elevado al número de orden del dígito (comenzando por el 0)

$$1110_2 = 1*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = 14_{10}$$

- De decimal a binario

La conversión se realiza dividiendo sucesivamente el número decimal entre 2, siendo los restos de las divisiones y el último cociente obtenido los dígitos del número en base binaria.

Ejemplo:

```
104 | 2
 04  52 | 2
  0  12  26 | 2
     0   06  13 | 2
         0   1   6 | 2
             0   3 | 2
                 1   1
```

El número en base dos se obtiene juntando el último cociente con todos los restos, comenzando por el de abajo: $104_{10} = 1101000_2$

Para pasar a binario la parte fraccionaria de un número decimal, se multiplica sucesivamente por la base, 2; indicando la parte entera de los resultados los dígitos del número en binario.

Ejemplo:

0.250 * 2 = 0.500

0.500 * 2 = 1.000

$0.250_{10} = 0.01_2$

- Aritmética binaria

Para realizar la suma, resta, multiplicación y división en base 2, se sigue la misma mecánica que con los números decimales, pero al existir sólo 2 cifras (1 y 0) resulta mucho más simple.
Las leyes fundamentales son:

SUMA:	0 + 0 = 0	RESTA:	0 - 0 = 0
(+)	0 + 1 = 1	(-)	0 - 1 = 1 (y me prestan 1)
	1 + 0 = 1		1 - 0 = 1
	1 + 1 = 0 (y me llevo 1)		1 - 1 = 0
MULT:	0 * 0 = 0	DIV:	0 / 0 = 0
(*)	0 * 1 = 0	(/)	0 / 1 = 0
	1 * 0 = 0		1 / 0 = infinito
	1 * 1 = 1		1 / 1 = 1

Ejemplos:

Suma

```
    0 1 1 0 1
    1 0 1 1 0
  + ---------
  1 0 0 0 1 1      (acarreo)
```

Resta

```
    1 0 0 1 0
      1 0 0 1
  - ---------
    0 1 0 0 1      (prestado)
```

Multiplicación

```
      1 0 0 1
        * 1 1
    ---------
      1 0 0 1
    1 0 0 1
  ---------
  1 1 0 1 1
```

División

```
1 0 0 0       1 1 0 1 | 1 1 1
  0 0 1 1 1             -------
        0 0 0
```

- Complementos

Llamamos complemento de un número a la diferencia entre la base en la que va expresado dicho número y él mismo.
Así por ejemplo, el complemento de 4 en base 10 es 6; el complemento de 4 en base 6 es 2; el complemento de 1 en base 2 es 1.
En aritmética binaria es muy común el uso de complementos, ya que como vamos a ver permiten realizar restas mediante sumas, y representar números negativos.

Complemento a 1:

El complemento a 1 de un número binario se obtiene cambiando todos los dígitos del número por sus opuestos, esto es, los unos se cambian por ceros, y viceversa.

Ej: el complemento a 1 de 100110 = 011001

Complemento a 2:

La manera más sencilla de conseguir el complemento a 2 de un número binario consiste en sumarle 1 al complemento a 1 de dicho número.

Ej: el complemento a 2 de 011011

0 1 1 0 1 1

1 0 0 1 0 0 (Complemento a 1)

+ 1 (Sumamos 1 a dicho complemento)

---

1 0 0 1 0 1 (Complemento a 2)

El complemento a 2 puede usarse para restar por medio de sumas (de hecho es así como se realizan las restas dentro del ordenador). Basta con sumar al minuendo el complemento a 2 del sustraendo. El acarreo del resultado se desprecia.

Ejemplo: Restamos $111_2 - 101_2$

-Hallamos el complemento a 2 del sustraendo (101)

1 0 1

0 1 0 (complemento a 1)

+ 1

---

0 1 1 (complemento a 2)

-Sumamos al minuendo el complemento a 2 del sustraendo

1 1 1

+ 0 1 1

---

1 0 1 0 (el acarreo se desprecia)

-1 1 1 - 1 0 1 = 0 1 0

La resta por suma de complemento no ocurre sólo con el sistema binario; en el sistema decimal ocurre exactamente igual

Ejemplo: Restamos 7 - 5

-Calculamos el complemento a 10 de 5

10 - 5 = 5

-Sumamos 7 el complemento a 10 de 5

```
   7
+  5
-----
   1  2        (el acarreo se desprecia)
```

El hacer estas restas con números en base decimal puede resultar ridículo, pero su uso es binario es básico, ya que así es como opera un ordenador para poder hacer una resta.

3. Sistema BCD

El sistema BCD es una forma algo peculiar de usar el sistema binario. En el sistema BCD cada dígito decimal se representa por medio de cuatro cifras binarias. Para pasar un número decimal a BCD, cada dígito decimal se transforma en binario, empleando para cada uno 4 bits.

Ejemplo: pasar a BCD 782_{10}

Decimal:	7	8	2
Binario:	0111	1000	0010

$782_{10} = 011110000010_{BCD}$

Hay instrucciones del 68008 que operan con datos en BCD

4. Sistema Hexadecimal

Es un sistema de numeración de base 16, cuyos dígitos simples son:0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Actualmente este sistema es muy usado al trabajar a nivel de lenguaje máquina, ya que todos los dígitos de este sistema se pueden representar con 4 bits binarios, con lo cual la conversión de cifras entre los dos sistemas es extremadamente simple.

Las equivalencias entre los 3 sistemas son:

Decimal	Hexadecimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001 0000
17	11	0001 0001
18	12	0001 0010

Para transformar un número hexadecimal a binario, se sustituye cada digito del número hexadecimal por el equivalente grupo de 4 bits binarios.

Ejemplo: Transformar $B5D_{16}$ en binario

HEXADECIMAL: B 5 D

BINARIO: 1011 0101 1101

$B5D_{16} = 101101011101_2$

Para convertir un número binario en hexadecimal se van agrupando los bits de 4 en 4, comenzando por la derecha, y se sustituye cada grupo de 4 bits por el equivalente digito hexadecimal.

Ejemplo: transformar 100101011010_2 en hexadecimal

BINARIO: 0010.0101.1010

HEXADEC.: 2 5 A

$100101011010_2 = 25A_{16}$

Para convertir un número hexadecimal en decimal se suman los valores que representa cada digito en funcial del lugar que ocupa, tal y como se hace con el sistema binario.

Ejemplo: pasar a decimal $52C_{16}$

$$52C = 5*16^2 + 2*16^1 + 12*16^0 = 1280 + 32 + 12 = 1324.$$

Para convertir un número decimal en hexadecimal se divide sucesivamente por 16, y cuando no se puedan continuar las divisiones se forma el número hexadecimal con el último cociente seguido de los restos sucesivos, desde el final hacia el principio.

Juan Palacio.

QUANTA

QUANTA se fundó en febrero de 1984 por su actual director Leon Heller y por su actual secretario Brian Pain. Particularmente considero una hazaña espectacular el que el Grupo se formase entre tres y cuatro meses antes de que Sinclair Research Ltd. comenzase a vender el tan anunciado y esperado QL.

El Grupo QUANTA empezó denominándose I.Q.L.U.G. que son las iniciales de Independent Q. L. Users' Group (Grupo de usuarios independientes del QL).

El ámbito de actuación del I.Q.L.U.G. es el Reino Unido y aunque su sede social está en Milton Keynes, el Grupo mantiene subgrupos por todo el Reino Unido a la vez que de vez en cuando organiza workshops en diversos lugares del mismo.

Desde el primer momento el I.Q.L.U.G. publica una revista para los socios, de periodicidad mensual, denominada QUANTA.

En el pasado año 1985 el Grupo tomó oficialmente el nombre de su revista para autodenominarse.

Aunque en un principio QUANTA solo tenia su propio significado (número cuántico o intervalo de un salto cuántico = cuantal, ahora además sus iniciales se corresponden con:

QL Users And Tinkers Association.

Como ya he dicho al comienzo, el ámbito de QUANTA es el Reino Unido pero además de tener socios en él, se puede decir que los tiene en todo el mundo.

Si alguien desea más información sobre QUANTA puede escribirme a mi o bien hacerlo directamente a Brian Pain a la dirección social del Grupo.

Ambos, Leon Heller y Brian Pain, son miembros de honor de QLave (C.E.I.U.Q.L.).

Por último, la dirección social de QUANTA es:

24 Oxford Street
Stony Stratford
Milton Keynes MK11 1JU
Tel: (0908) 564271
(United Kingdom)

S.Q.Y.

TRUCOS RUTINAS ...

En este número comenzamos a presentar una serie de rutinas cortas en ensamblador que consideramos os pueden interesar.

En este y en sucesivos listados intentaremos daros y esperamos que aquellos que nos enviéis rutinas lo hagáis también, junto con la rutina en ensamblador el listado basic con la equivalencia de éste en DATA.

Programa cargador:

```
10 C=RESERVE(70)
20 FOR I=0 TO 57
30  READ X:POKE I+C,X
40 END FOR I
50 EXEC MOV2_INFOR,C,70,10
60 DATA 112,0,78,65,71,250,0,50,36,130,114,255,118,2,65,250,0,34,112,1,78,66,116
       ,10,118,255,67,250,0,28,112,7,78,67,112,2,78,66,114,255,118,0,112,5,78,65,
       112,0,78,117,0,4,83,67,92,95,0,10
```

Este programa permite obtener la versión del QDOS que esté corriendo en nuestro ordenador, algo necesario de conocer ya que existen versiones antiguas que no aceptan algunas "llamadas" ya implantadas en las nuevas versiones.

El programa tiene cuatro bloques básicos y ha sido realizado con el programa ensamblador que proporciona la casa METACOMCO. Estos cuatro bloques son etiquetas de llamadas al QDOS, Macros definidas, el código ejecutable en sí, zona de memoria de datos.

El primer bloque se refiere al valor que se debe pasar en el registro de datos D0 al realizar el Trap (orden en ensamblador que se refiere a una excepción y que es la forma habitual de trabajo del QDOS) correspondiente.

He usado dos Macros, algo bastante recomendable si se dispone de un ensamblador que lo permita. La primera nos sirve para pasar las llamadas al QDOS (el parámetro en D0 y el número de Trap que corresponda). La segunda se encarga de finalizar el programa y devolvernos al BASIC.

El programa comienza con una llamada para obtener la información del la versión esto se hace con un Trap 1 y MT_INF en D0 el resultado se saca en D2 que guardamos en la zona de memoria que hemos reservado para ello, la zona etiquetada como MENSAJE donde con 4 bytes habría bastado en lugar de 10.

Ahora para sacar estos datos podemos hacerlo por pantalla o por impresora nosotros lo haremos por pantalla de ahí que DISPOSITIVO sea 'SCR_'. Abrimos un canal para efectuar la salida para ello usamos un Trap 2 con IO_OPEN en D0, en D1 el número de job que se trate (en este caso -1 para indicar el actual), en D3 un 2 para indicar que es nuevo, y A0 apunta al DISPOSITIVO, que aunque aquí sea SCR_ podríamos colocar SER1 para impresora por ejemplo.

Para sacar los datos se usa un Trap 3 con IO_SSTRG en D0, hay que especificar el número de octetos a mandar en D2 y el tiempo de espera en D3, A1 debe de apuntar a la zona de memoria que contiene los octetos a mandar.

Para finalizar se invoca la Macro FINAL que cierra el canal abierto con un Trap 2 conteniendo IO_CLOSE en D0. Posteriormente forzamos la clausura del job y como éste es activo lo hacemos con un Trap 1 conteniendo D0 MT_FRJOB, en D1 está el número de job y en D3 el código de error.

Isidro Asín.

```
*
*Etiquetas referentes al GDOS
*
MT_INF          EQU             0 Obtener información de la versión del GDOS
IO_OPEN         EQU             1 Abrir canales
IO_CLOSE        EQU             2 Cerrar canales
IO_SSTRG        EQU             7 Presenta datos por un canal de salida
MT_FRJOB        EQU             5 Cancela un job
*
*Macros
*
  GDOS          MACRO                   Macro de asignación de Traps
                MOVEQ           #\1,D0 Parámetro en D0
                TRAP            #\2    Parámetro del Trap
                ENDM
  FINAL         MACRO                   Macro de cancelación de canales y job
                GDOS            IO_CLOSE,2 Cierra canal
                MOVEQ           #-1,D1     del job actual
                MOVEQ           #0,D3      Código de error
                GDOS            MT_FRJOB,1 Llamada GDOS
                MOVEQ           #0,D0      Sin errores
                RTS
                ENDM
*
*Programa principal
*
                GDOS            MT_INF,1   Obtengo información
                LEA.L           MENSAJE,A3 Apunto A3 a la zona donde guardo infor.
                MOVE.L          D2,(A3)    Llevo allí el num. de versión
                MOVEQ           #-1,D1     Job actual
                MOVEQ           #2,D3      Nuevo fichero
                LEA.L           DISPOSITIVO,A0 Tipo de dispositivo asociado
                GDOS            IO_OPEN,2  Llamada al GDOS
                MOVEQ           #10,D2     Mando octetos de más
                MOVEQ           #-1,D3     Tiempo de espera infinito
                LEA.L           MENSAJE,A1 apunto a la base de la memoria
                GDOS            IO_SSTRG,3 Llamada al GDOS
*
                FINAL

DISPOSITIVO     DC.W            4
                DC.B            'SCR_'     Dispositivo de salida es la pantalla
MENSAJE         DC.B            10         Base de la memoria

                END
```